terry lyu 435091

Q1
Part a b c

```
data = LOAD 'dualcore/ad_data[1-2]' AS (campaign_id:chararray,
            date:chararray, time:chararray,
           keyword:chararray, display_site:chararray,
            placement:chararray, was_clicked:int, cpc:int);

group_site = GROUP data BY display_site;

groupby_site = FOREACH group_site {
clicked = FILTER data BY was_clicked == 1;
num_clicked = COUNT(clicked);
total = COUNT(data);
ctr = num_clicked*100.0/total;
GENERATE group, ctr AS ctr;

}



ascsorted = ORDER groupby_site BY ctr ASC;


lowest = LIMIT ascsorted 3;

get_keyword = GROUP data BY keyword;

group_keyword = FOREACH get_keyword {
clicked = FILTER data BY was_clicked == 1;
num_clicked = COUNT(clicked);
total = COUNT(data);
ctr = (100.0*num_clicked)/total;
GENERATE group, ctr AS ctr;
}

keyword_desc_sorted = ORDER group_keyword BY ctr DESC;
threehighkeyword = LIMIT keyword_desc_sorted 3;
DUMP lowest;
DUMP threehighkeyword;
```

Three site with lowest click rate is (bassoonenthusiast.example.com, 1.000741289844329)
(grillingtips.example.com,1.7343173431734318)
(butterworld.example.com,1.90032269630692),

Three sites with highest click rate is

(PRESENT,6.449976753032672)
(BARGAIN,3.7029166445306276)
(BYTEWEASEL,3.5706739911261356)

Q2

(2013-02,76170)
(2013-03,84549)
(2013-04,87853)
(2013-05,115038), yes it does show significant increase in sales.

```
data = LOAD 'dualcore/orders' AS (order_id:int,
          cust_id:int,
          order_dtm:chararray);


filtered_order = FILTER data by order_dtm matches '^2013-0[2345]-\\d{2}\\s.*$';

order_ym = FOREACH filtered_order GENERATE SUBSTRING(order_dtm, 0, 7) as ym;



order_pm = GROUP order_ym BY ym;
order_m = FOREACH order_pm GENERATE group, COUNT(order_ym.ym);

DUMP order_m;
```

q3
we should use the orders dataset for this questions. before writing any code, i think i should dump the dataset to see the pattern of the data

There are filter statement after loading statement. When the dataset is extremely large, filtering out the unnecessary datas can limit the amount of data that needs to be processed.

```
orders = LOAD 'dualcore/orders' AS (order_id:int,
          cust_id:int,
          order_dtm:chararray);

details = LOAD 'dualcore/order_details' AS (order_id:int,
          prod_id:int);


recent = FILTER orders BY order_dtm matches '^2013-0[2345]-.*$';


tablets = FILTER details BY prod_id == 1274348;



join_by_orderid = JOIN recent BY order_id, tablets BY order_id;

only_month = FOREACH join_by_orderid GENERATE SUBSTRING(recent::order_dtm, 0, 7) as grouped_month;

group_by_month = GROUP only_month BY grouped_month;
count_order= FOREACH group_by_month GENERATE group,
```

```
COUNT(only_month.grouped_month);



DUMP count_order;

(2013-02,3598)
(2013-03,3904)
(2013-04,4134)
(2013-05,49514)


the data does show a significant increase while the campaign is active.

q4

we need to use the order dataset, the detail dataset, the products dataset.
orders = LOAD '/dualcore/orders' AS (order_id:int,
          cust_id:int,
          order_dtm:chararray);

details = LOAD '/dualcore/order_details' AS (order_id:int,
          prod_id:int);

products = LOAD '/dualcore/products' AS (prod_id:int,
          brand:chararray,
          name:chararray,
          price:int,
          cost:int,
          shipping_wt:int);



Part 23

-- load the data sets
orders = LOAD 'dualcore/orders' AS (order_id:int,
          cust_id:int,
          order_dtm:chararray);

details = LOAD 'dualcore/order_details' AS (order_id:int,
          prod_id:int);

products = LOAD 'dualcore/products' AS (prod_id:int,
          brand:chararray,
          name:chararray,
          price:int,
          cost:int,
          shipping_wt:int);



order_from_2012 = FILTER orders BY order_dtm matches '^2012.*$';


order1 = GROUP orders BY cust_id;
order2 = FOREACH order1 GENERATE group, COUNT(orders) AS ordercount;
```

```
order3= FILTER order2 BY ordercount >= 5;

order4= JOIN order3 BY group, order_from_2012 BY cust_id;
order_detail = JOIN order4 BY order_from_2012::order_id, details BY order_id;
order_product = JOIN order_detail BY details::prod_id, products BY prod_id;
order_price = FOREACH order_product GENERATE order3::group AS cust_id,
products::price AS price;
customer1 = GROUP order_price BY cust_id;
customerfinal = FOREACH customer1 GENERATE group, SUM(order_price.price) AS total;
SPLIT customerfinal INTO
platinum IF total >= 10000,
gold IF total >= 5000 AND total < 10000,
silver IF total >= 2500 AND total < 5000;
STORE platinum INTO 'dualcore/loyalty/platinum';
STORE gold INTO 'dualcore/loyalty/gold';
STORE silver INTO ‘dualcore/loyalty/silver';
```

there are 111655 customers in platinum, 10976 in gold, 5241 in silver.